Optimizing Complex Extraction Programs over Evolving Text Data

Fei Chen[1], Byron Gao[2], AnHai Doan[1], Jun Yang[3], Raghu Ramakrishnan[4]

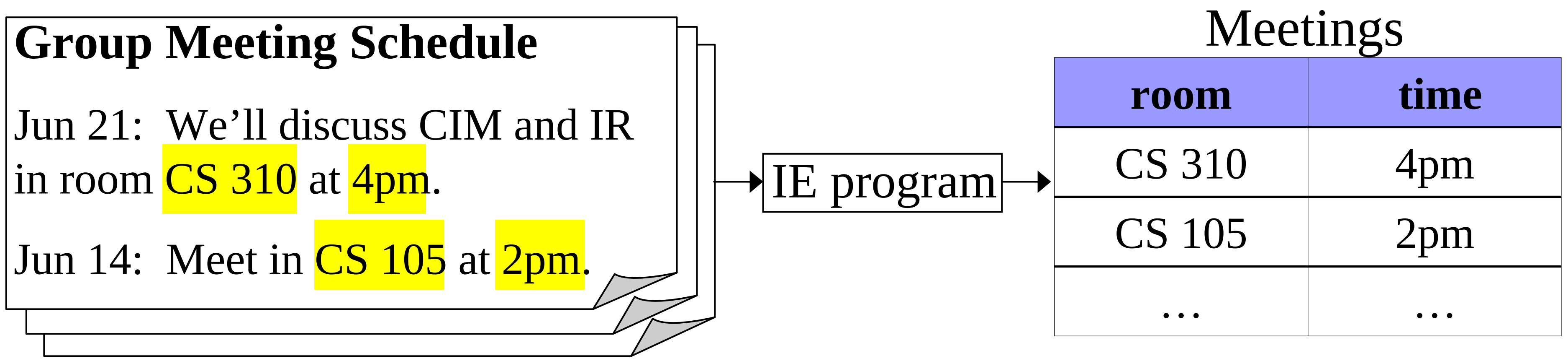
[1]University of Wisconsin-Madison

[2]Texas State University-San Marcos

[3]Duke University

[4]Yahoo! Research

Information Extraction (IE)



Group Meeting Schedule

Jun 21: We'll discuss CIM and IR in room CS 310 at 4pm.

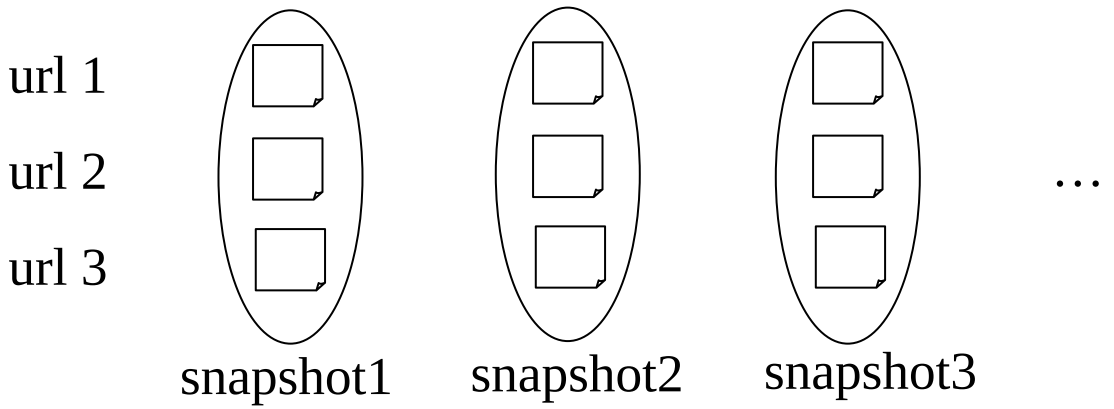
Jun 14: Meet in CS 105 at 2pm.

Meetings

room	time
CS 310	4pm
CS 105	2pm
…	…

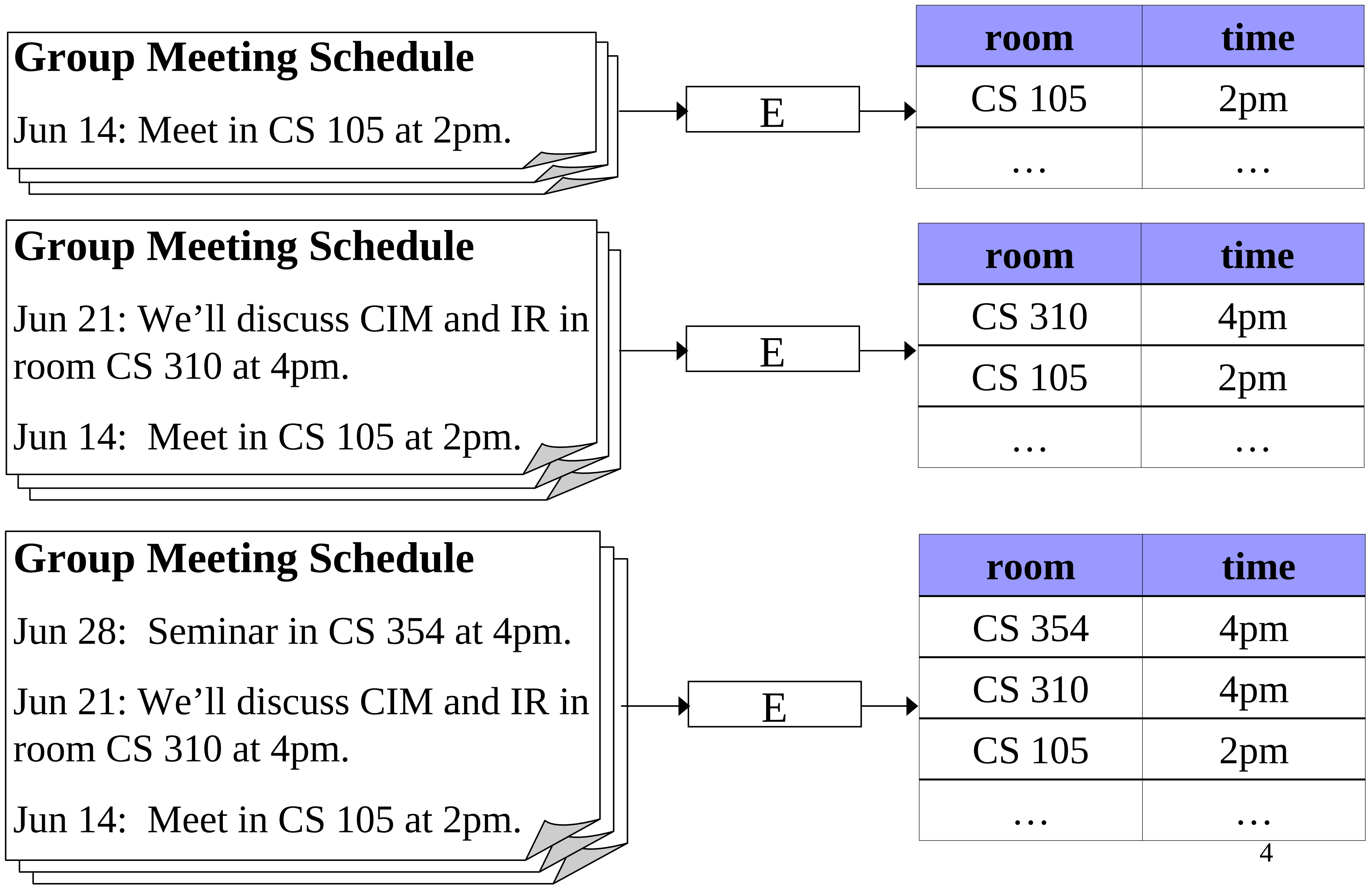
- **Many solutions in database/Web/AI communities with significant progress**
- **But most solutions have considered only static text corpora**

Evolving Text Corpora Are Pervasive



- **Impliance @ IBM**
  - find the latest information from enterprise intranets
- **IWP@Univ. of Washington and YAGO@MPI**
  - keep extracted knowledge consistent with the Wikipedia pages
- **DBLife@Univ. of Wisconsin**
  - monitor community information

IE over Evolving Text Data



Group Meeting Schedule

Jun 14: Meet in CS 105 at 2pm.

room	time
CS 105	2pm
…	…

Group Meeting Schedule

Jun 21: We’ll discuss CIM and IR in room CS 310 at 4pm.

Jun 14: Meet in CS 105 at 2pm.

room	time
CS 310	4pm
CS 105	2pm
…	…

Group Meeting Schedule

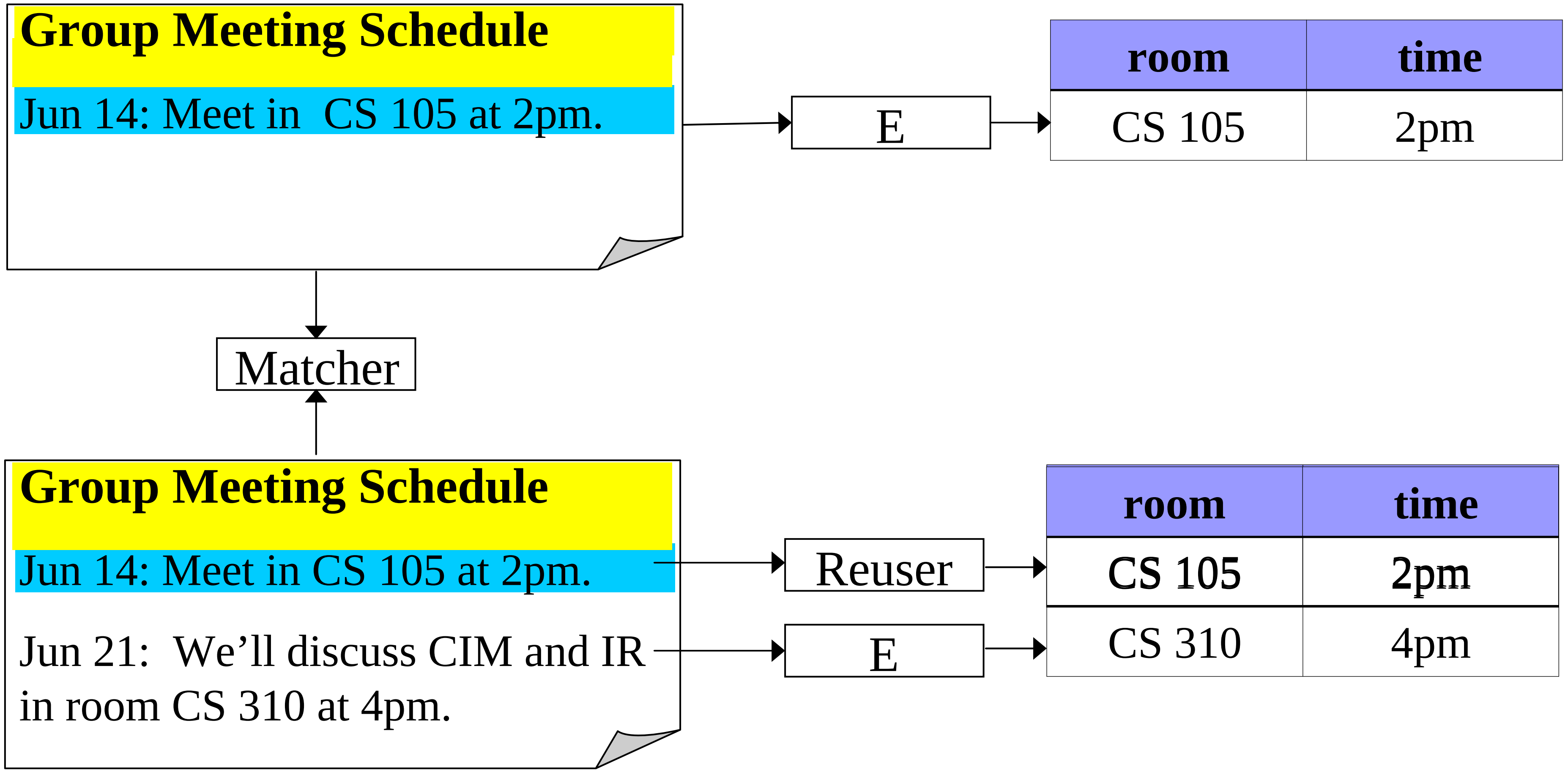
Jun 28: Seminar in CS 354 at 4pm.

Jun 21: We’ll discuss CIM and IR in room CS 310 at 4pm.

Jun 14: Meet in CS 105 at 2pm.

room	time
CS 354	4pm
CS 310	4pm
CS 105	2pm
…	…

Cyclex[ICDE08]: Match, Reuse and Extract


Group Meeting Schedule
Jun 14: Meet in CS 105 at 2pm.
E
room
time
CS 105
2pm
Matcher
Group Meeting Schedule
Jun 14: Meet in CS 105 at 2pm.
Jun 21: We'll discuss CIM and IR in room CS 310 at 4pm.
Reuser
E
room
time
CS 105
2pm
CS 310
4pm

Cyclex[ICDE08]: Properties of Extractors for Correct Reuse

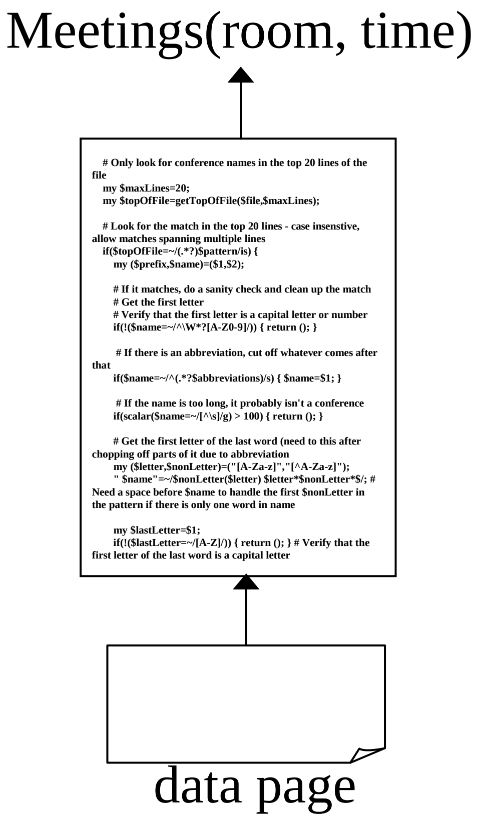
- **Scope: max length of any mention extracted by E**

- **Context: length of “text windows” surrounding a mention**
  - E only exams the text windows to extract the mention

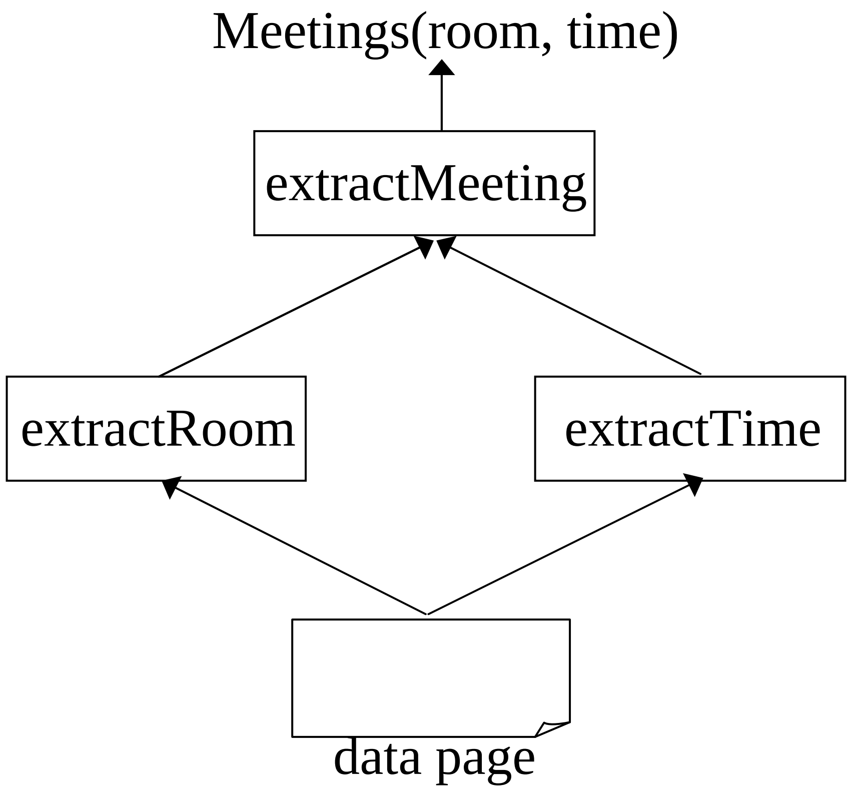
Example : E extracts telephone numbers using regular expression

$$\text{“}\underbrace{\textbf{be reached at}}_{\text{context = 14 chars}}\ \underbrace{\textbf{\textbackslash d\{7\}}}_{\text{scope = 7 chars}}\text{”}$$

Limitations of Cyclex[ICDE08]

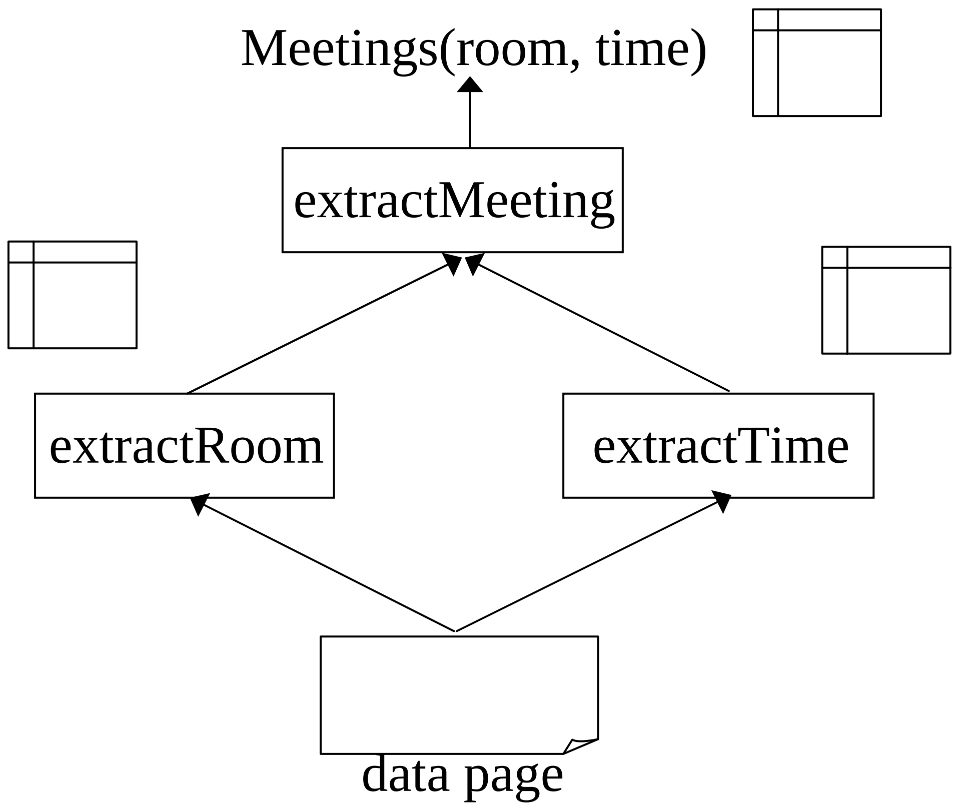


```
   # Only look for conference names in the top 20 lines of the
file
   my $maxLines=20;
   my $topOfFile=getTopOfFile($file,$maxLines);

   # Look for the match in the top 20 lines - case insenstive,
allow matches spanning multiple lines
   if($topOfFile=~/(.*?)$pattern/is) {
      my ($prefix,$name)=($1,$2);

      # If it matches, do a sanity check and clean up the match
      # Get the first letter
      # Verify that the first letter is a capital letter or number
      if(!($name=~/^\W*?[A-Z0-9]/)) { return (); }

       # If there is an abbreviation, cut off whatever comes after
that
      if($name=~/^(.*?$abbreviations)/s) { $name=$1; }

       # If the name is too long, it probably isn't a conference
      if(scalar($name=~/[^\s]/g) > 100) { return (); }

      # Get the first letter of the last word (need to this after
chopping off parts of it due to abbreviation
      my ($letter,$nonLetter)=("[A-Za-z]","[^A-Za-z]");
      " $name"=~/$nonLetter($letter) $letter*$nonLetter*$/; #
Need a space before $name to handle the first $nonLetter in
the pattern if there is only one word in name

      my $lastLetter=$1;
      if(!($lastLetter=~/[A-Z]/)) { return (); } # Verify that the
first letter of the last word is a capital letter
```



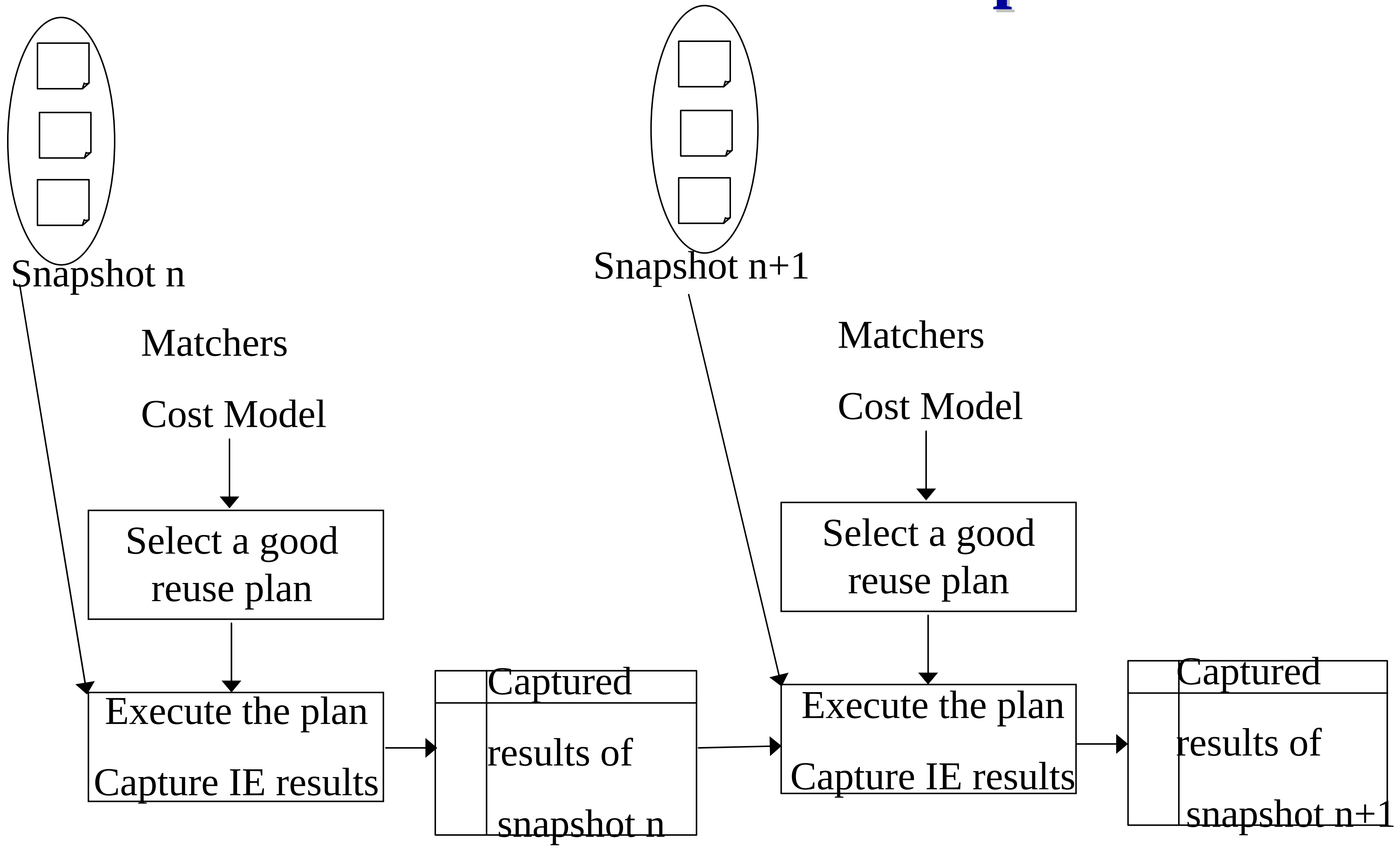
- **Cyclex treats an IE program as a blackbox**
- **Real-world IE programs are complex**
  - Avatar: 25+ blackboxes
  - DBLife: 45+ blackboxes

Delex: Decompose and Recycle



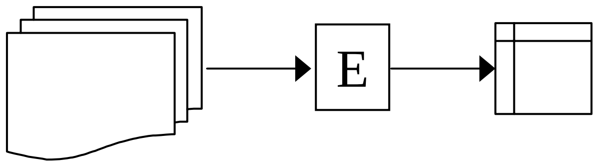
- Exploit the composition nature of IE programs
- Delex cuts the runtime of Cyclex by 50-71%

Delex on Consecutive Snapshots



A Baseline Solution of Capturing Results for Multi-Blackbox IE Programs

- **Capturing results in Cyclex**

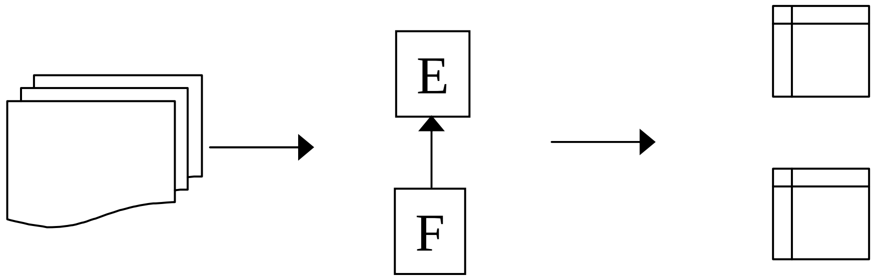


{(mention extracted,

docID,

positions within the doc)}

- **A baseline solution: applying the solution of Cyclex to each blackbox**

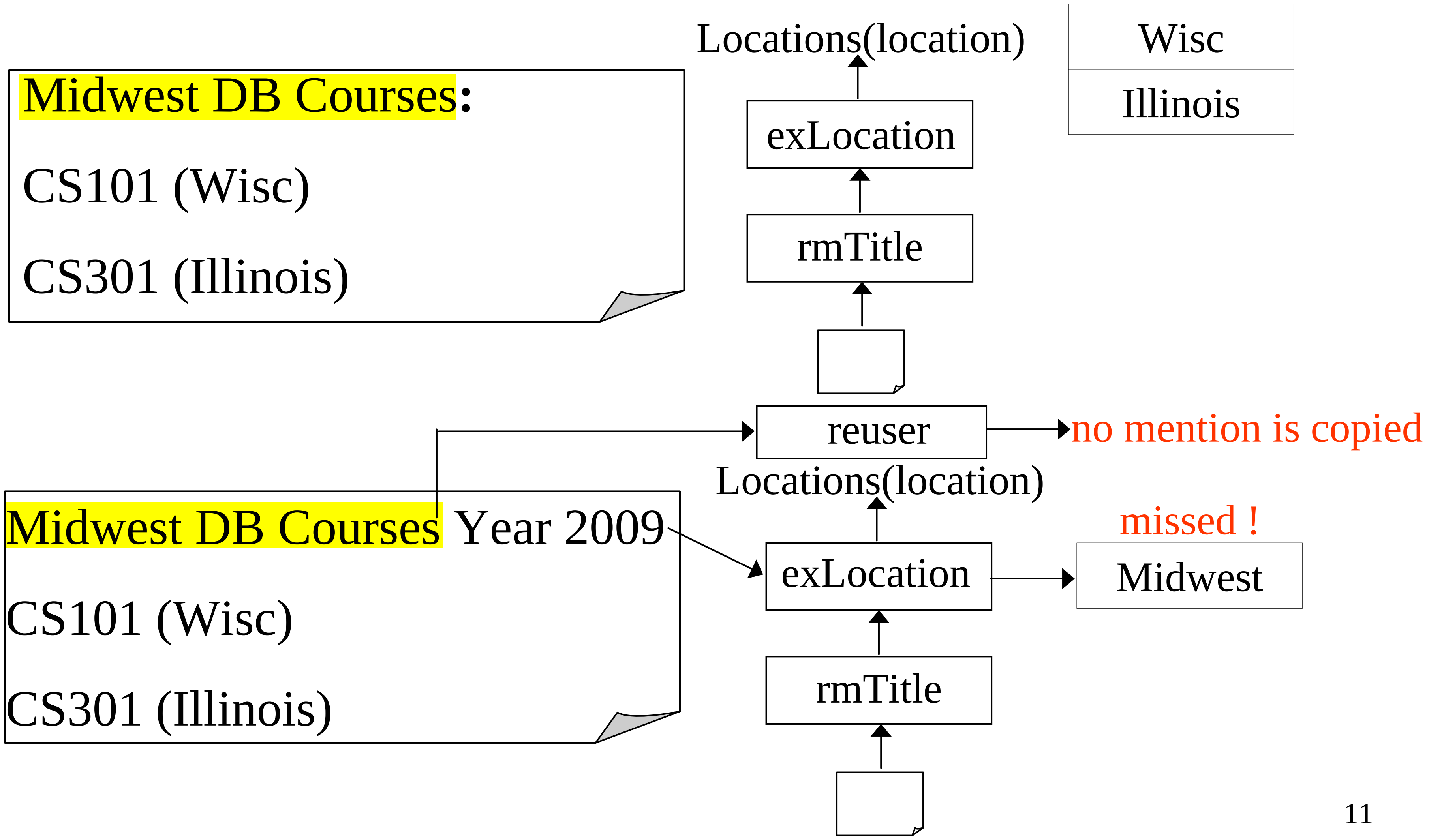


{(mention extracted,

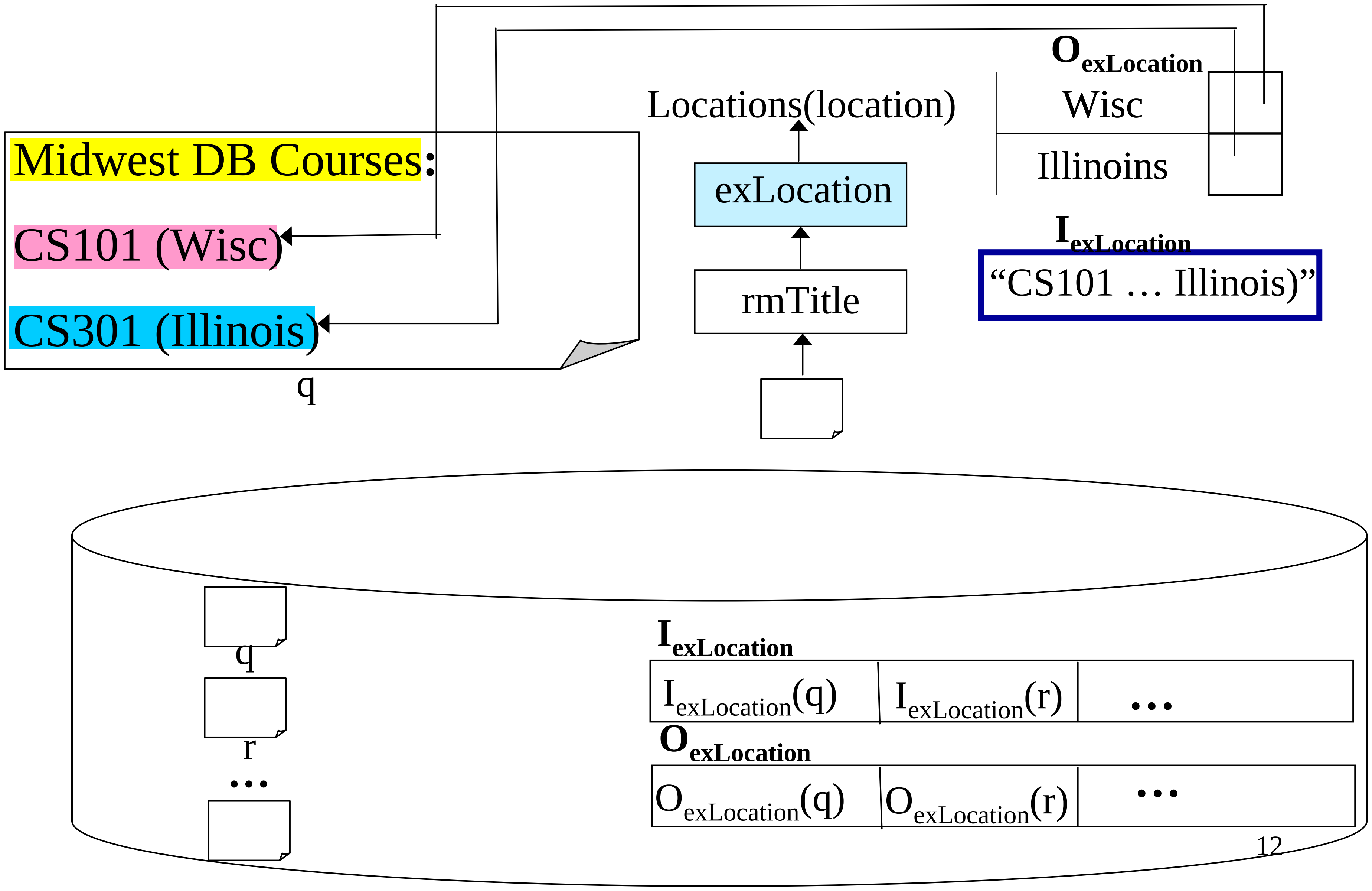
docID,

positions within the doc)

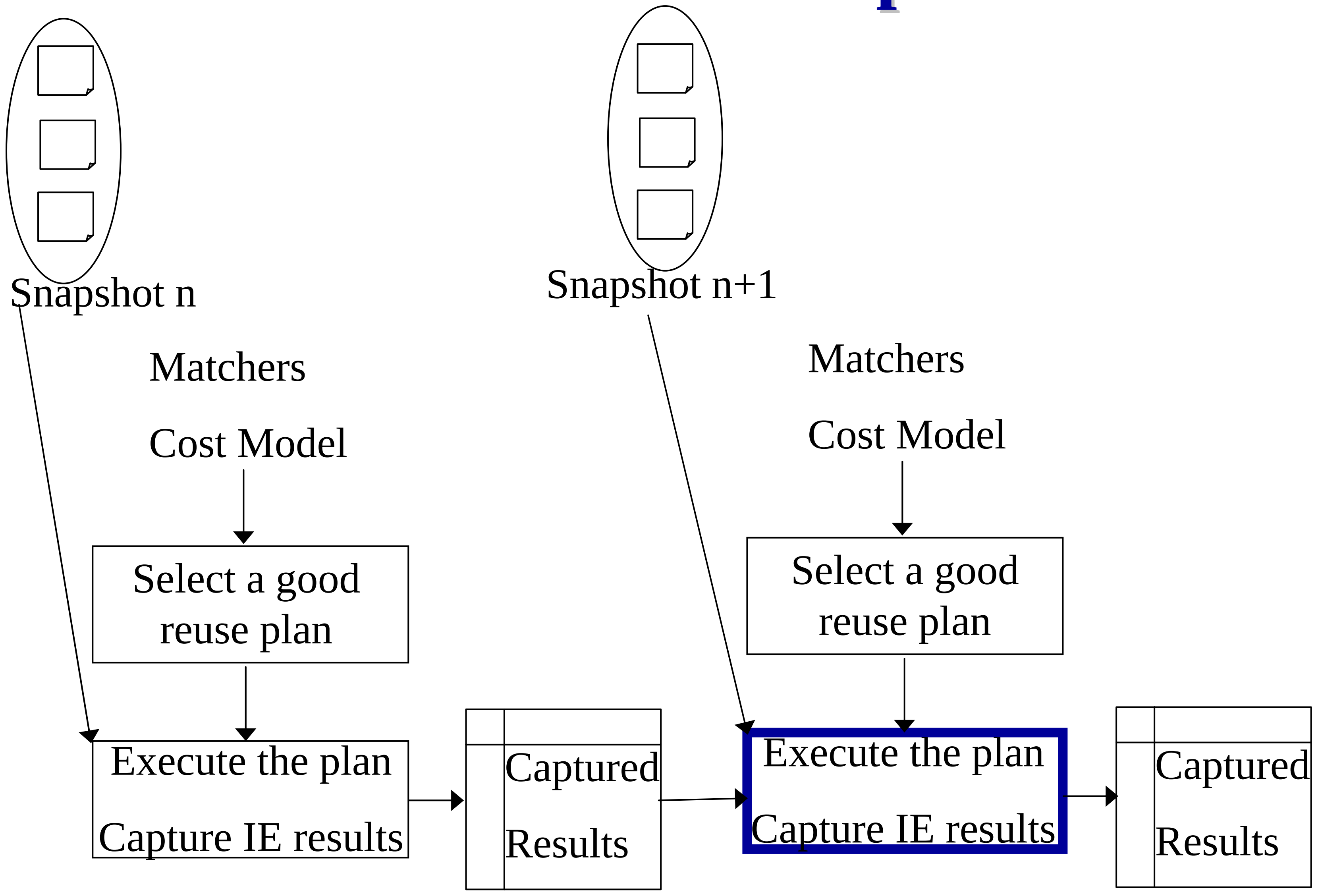
The Baseline Solution May Miss Mentions



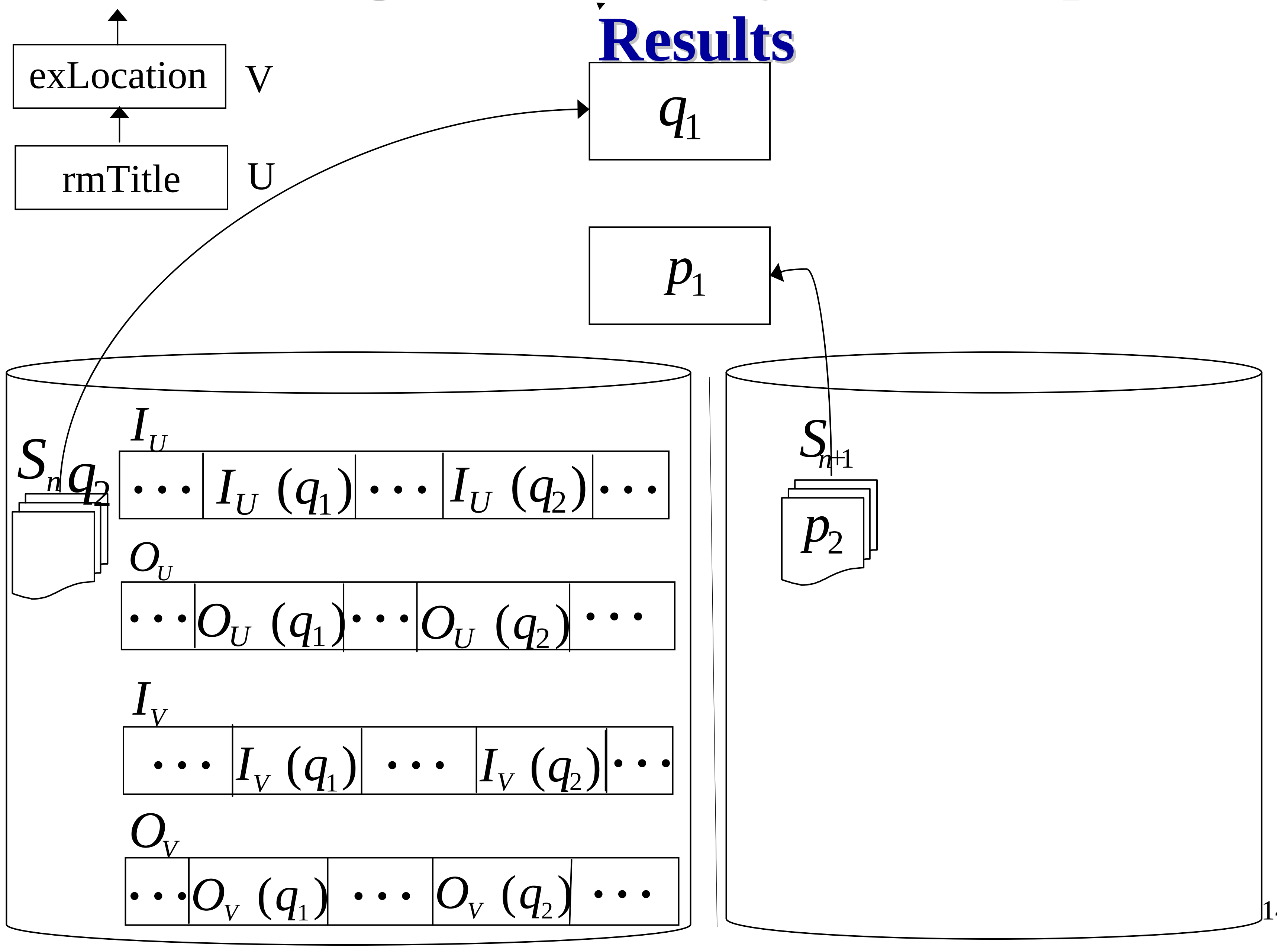
Capture and Store IE Results in Delex



Delex on Consecutive Snapshots

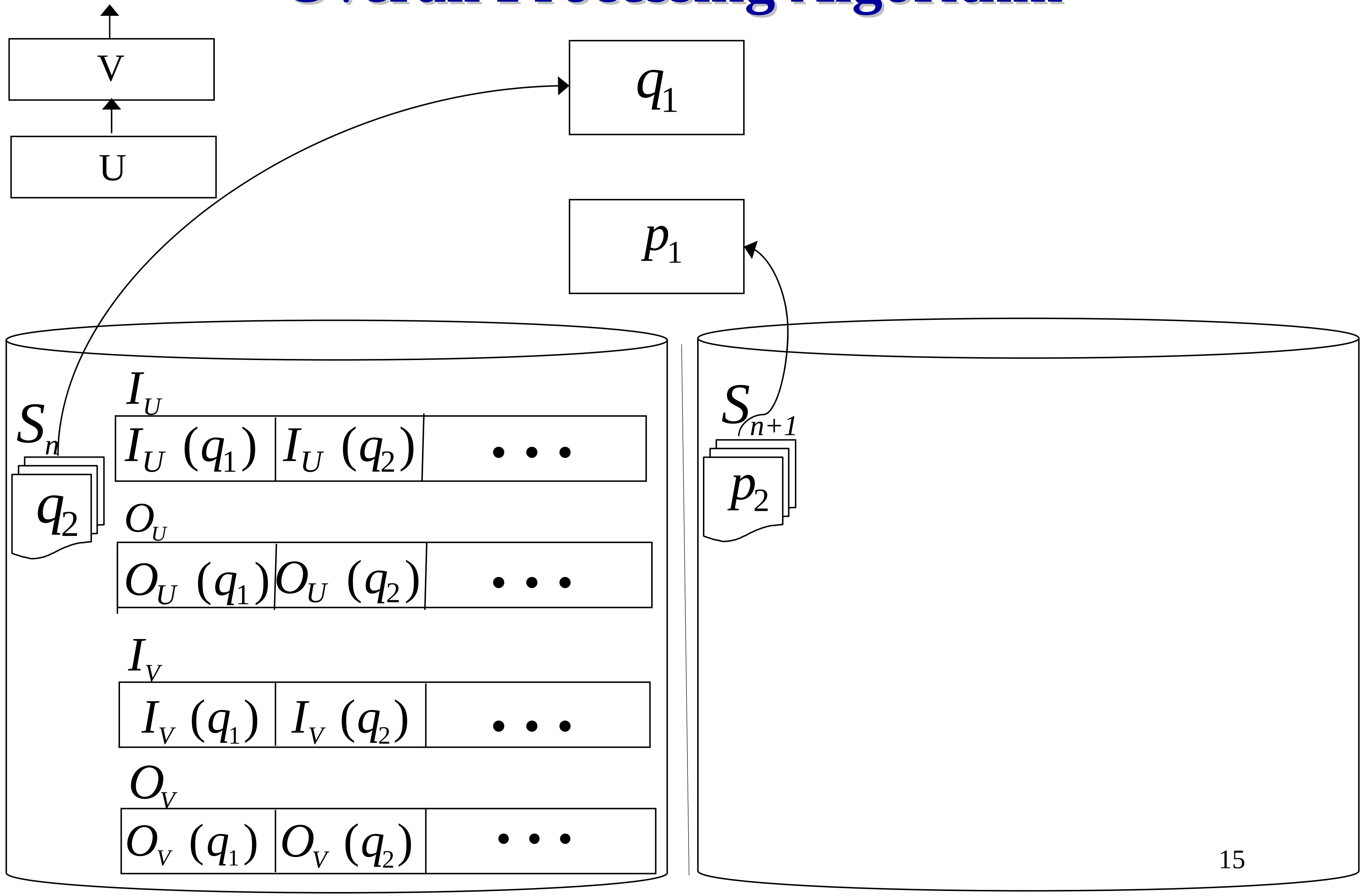


Challenge in Efficiently Reuse Captured Results

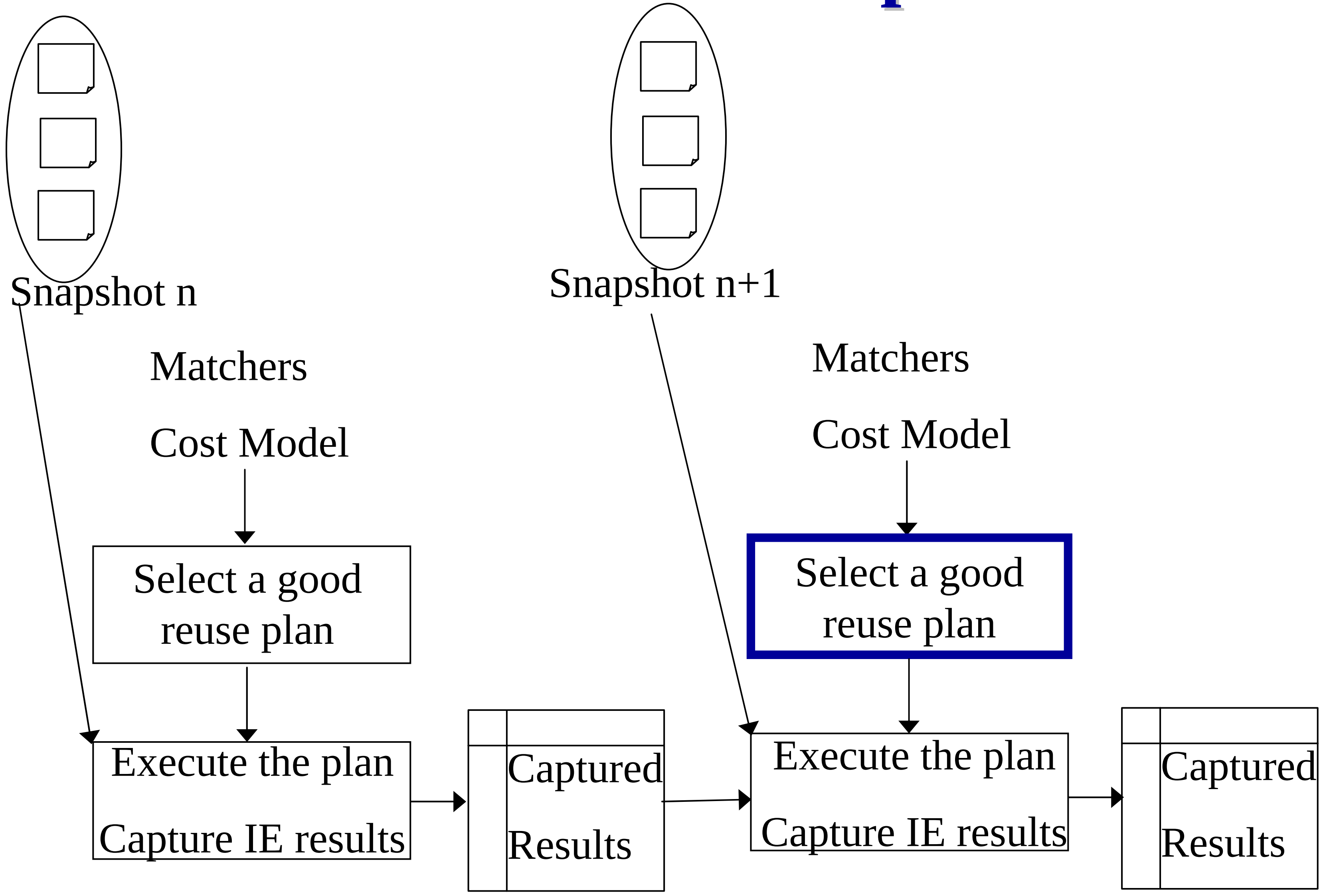


exLocation
V
rmTitle
U
q_1
p_1
S_n
q_2
I_U
$\cdots$ $I_U(q_1)$ $\cdots$ $I_U(q_2)$ $\cdots$
O_U
$\cdots$ $O_U(q_1)$ $\cdots$ $O_U(q_2)$ $\cdots$
I_V
$\cdots$ $I_V(q_1)$ $\cdots$ $I_V(q_2)$ $\cdots$
O_V
$\cdots$ $O_V(q_1)$ $\cdots$ $O_V(q_2)$ $\cdots$
S_{n+1}
p_2

Overall Processing Algorithm



Delex on Consecutive Snapshots



Find a Good Reuse Plan

- **Plan space**
  - assign a matcher to each IE blackbox
  - # of plans is exponential in # of IE blackboxes

- **Use a text-specific cost model to estimate the completion time of each plan**

- **Searching for good plans**
  - optimization is not "decomposable"
  - a greedy solution that efficiently finds a good plan

Experiment Setup

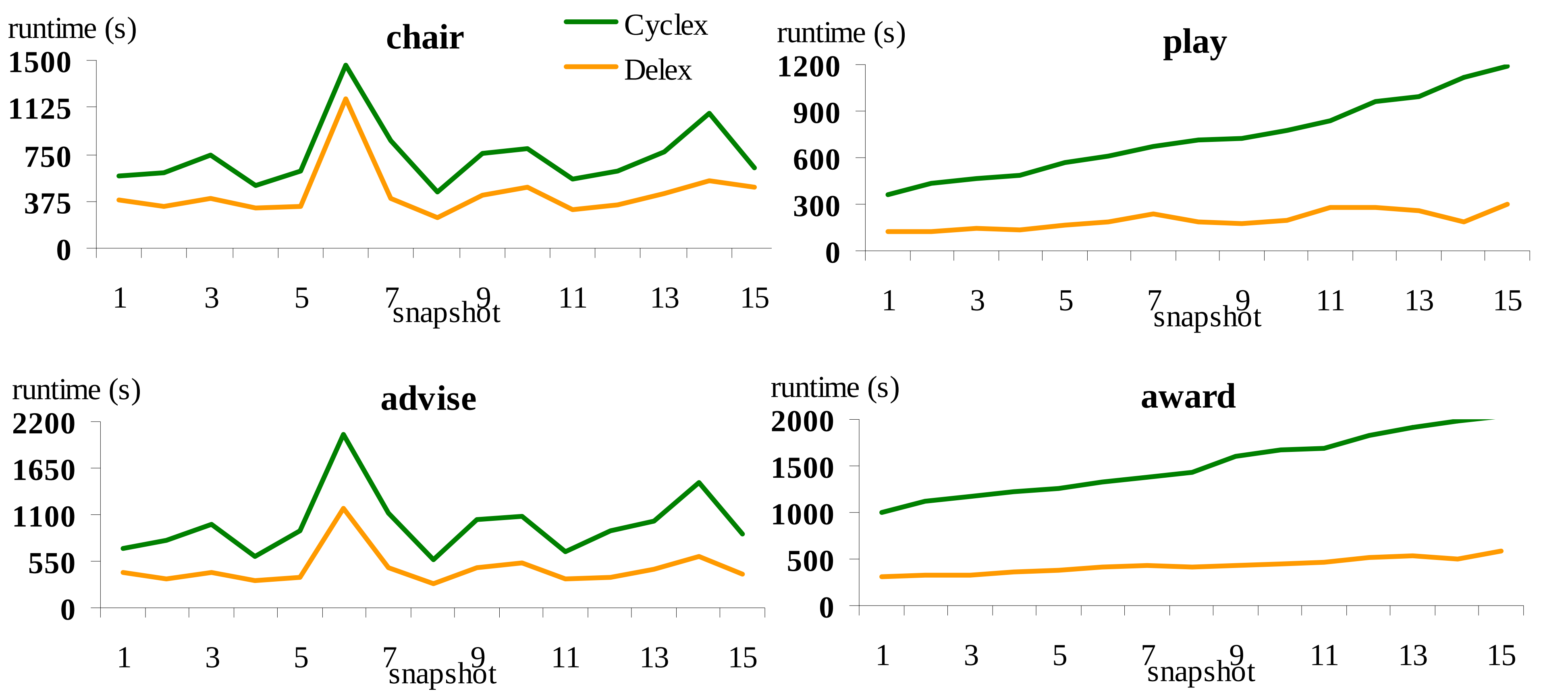
- **Datasets**

Data Sets	**DBLife**	**Wikipedia**
# Snapshots	15	15
Time between snapshots	2 days	21 days
Avg # Page per Snapshot	10155	3038
Avg Size per Snapshot	180M	35M

- **IE Programs : Rule-based and Learning-based IE Programs**

	DBLife (Rule-based)			**Wikipedia (Rule-based)**			**Wikipedia (Learning-based)**
	talk	chair	advise	blockbuster	play	award	actor
# of IE “Blackboxes”	1	3	5	2	4	6	5

Runtime Comparison



- **Delex drastically cuts runtime of cyclex by 50-71%**

(See paper for more experiments)

Related Work

- **IE over evolving text data**
  - [Doan et al, ICDE-08]
  - only considers a single IE blackbox

- **Optimizing IE programs**
  - [Gravano et al, SIGMOD-06] [Gravano et al, ICDE-07] [Doan et al, VLDB-07] [Reiss et al, ICDE-08]
  - only consider static text corpora

- **Incremental View Maintenance**
  - [Gupta&Mumick][&Widom et al, SIGMOD-95][Garcia-Molina&Widom et al, VLDB-91]…
  - only consider relational operators with well defined semantics
  - assume that changes to the inputs are readily available

Conclusion and Future Work

- **First in-depth solution to optimizing complex IE over evolving text**

- **Defined challenges and provided initial solutions**
  - capture intermediate IE results for correct reuse
  - efficiently coordinate matching, extraction, and copying for multiple IE blackboxes
  - cost-based decisions in choosing a good reuse plan

- **Future work**
  - reuse across URLs
  - handle extractors that extract mentions across multiple pages